

From reviewer #1, "*Because the paper focuses on determination of ramp dimensions using VA method, this Ms must.. say something about the ramp period. Otherwise, a reader has the feeling that the Ms is incomplete.... the present Ms must to add a sentence about the ramp period. In fact, I do not understand why it was not included because after determination of the ramp amplitude and the time lag that peaks  $S3r/r$  the calculation of the ramp period is straightforward using VA. Thus, no extra figures or calculations must be implemented.*" I have added a more complete description of the SR calculation of flux, including the determination of the ramp period ( $\tau$ ) as it relates to computation efficiency. I have also reordered some of the method description to be more pertinent to the surface renewal method in particular.

Also from Reviewer #1, "*Because the authors mentioned that there is a companion paper where they compared (I guess) SR and EC fluxes, one may hesitate the latter issue was addressed in the forthcoming Ms.*" I have clarified that the experiment alluded to in the manuscript does calibrate surface renewal flux on the basis of fluxes calculated via eddy covariance.

From Reviewer #2, "*I am eager to try the supplemental code over my own data.*" I have updated the manuscript with a declaration of data availability (with a DOI), and listed the URL from which data and complete code can be downloaded. "*Some clarification will be necessary in parts of the manuscript.*" I have made some attempt to identify some ambiguous references and clarify these, expecting that the discussion forum will raise additional needed clarifications. "*Also, very important comment on possibility to use this computationally efficient approach in dataloggers programming for online flux computation is needed in more detailed discussion.*". While this is a research interest that will be pursued, no attempt has been made to date to implement such computational approaches in a data logger programming language. Nonetheless, it should be straightforward to program the most relevant aspects of these techniques in languages that are pre-compiled with methods such as the FFT (for example, Campbell Scientific loggers using CRBasic). The authors will attempt to address this in the discussion forum, as it is not a task that has actually been completed to date.

*GENERAL COMMENTS:*

The authors gratefully acknowledge the reviewer's comments, which have led to substantial changes in the manuscript, which was initially revised in May 2017, and again recently in response to additional review. Overall, I have attempted to improve clarity of writing; specifically, the hypotheses and goals of the manuscript have been clarified, and the methods are more clearly delineated with appropriate and up to date citations. The brief conclusion has been changed to be more appropriate, in particular to describe additional results which have been included in response to additional reviewer requests. The most significant of these was expanding on actual implementation of despiking via convolution, using a phase space criteria for quantitative spike identification. I have also attempted to clarify those parts of the analysis which are novel in implementation, within the context of previous work in programmatic algorithms.

Specific comments are addressed in the supplementary notes, attached.

*SPECIFIC COMMENTS:*

*P1L9: "20 Hz+" I suggest changing to "10+ Hz" since there are articles published on successful SR application over data collected at 10 Hz frequency. Similarly, it is mentioned 10+ Hz in the introduction section.*

Changed as requested. Further clarifications later in the text as to different sampling frequencies and the corresponding cost in data storage and computational efficiency.

*P1L13: Please add, "computational" before the word "efficiency"*

Added as requested.

*P1L14: Please, be more specific, i.e. avoid saying "Programming techniques such as these", it is still not clear in the abstract what you are referring to...*

This section has been re-written to be more concrete and an appropriate summary.

*P1L19: Please add word "possible" between "determinations" and "using".*

Added as requested.

*P2L50: Please clarify. It is said in this line that the convolution is used for structure functions computation, while later is said the convolution is used to despike the data. It seems that the convolution is used for both and should be explained.*

Added clarification here as requested.

*P2L51: Replace "simplifying" with "to simplify"*

Replaced as requested.

*P2L53: Please, start the Methods differently... it is always better for the reader to be more specific.*

Changed with the intent to be more specific for the reader.

*P2L59: Add, "the method" before "implementation"*

Added as requested.

*P4L105: Please change the word “approach” to “program” or “program run” for clarity.*  
Changed as requested.

*P4L111-113: Please rewrite for the clarity.*  
Changed as requested.

*P4L114: Please replace “identical” by “from the same 20 Hz dataset”.*  
Replaced as requested.

*P4L131 and L133: Should “N-1” be actually “N-r”?*  
Yes, thank you for catching this error in the text.

*P4L133: Should “T(t)” be “T(i)” in accordance to Eq.1. and for clarity?*  
No, this sentence refers to the time series rather than the discrete samples, but I agree it was unclear due to proximity of the terms. It has been changed for clarity.

*P5L137: “sweeps and ejections” of what? It should be clarified adding more description of the surface renewal method background.*  
Added as requested.

*P5L138: Please add “ramps in the temperature signal caused by” between “geometry of” and “coherent structures”*  
Added as requested.

*P5L140 and L141: Please change to be clearer that the detection of the structure functions in the scalar signals improved.*  
Added discussion of the improved detection and concurrent increase in statistical robustness with larger sample sizes (higher frequency measurement)

*P5L144: Explain under which conditions.*  
Added as requested. Other reviewers noted the ambiguity of this statement so a short discussion of flux direction and atmospheric stability was added.

*P5L146: Please replace “to” by “for” if I understand well. In addition, explain why are the periods of 8 hours collected, and not continuously for two months. What challenges did you find? Was it only data collection for the unstable conditions?*  
This language was too abbreviated in the text, and has been updated to reflect that I am describing several different experiments. The description now indicates that this data was used to test the methods with different performance criteria- i.e. data of different sizes, experiment durations, and captured at different frequencies.

*P5L147-L150: Please explain how this analysis is useful. Are fluxes better to be calculated for 3 minute periods?*  
The introduction and discussion have been expanded to indicate why short duration fluxes are under consideration.

*P5L163: What “total data size” means? It works with the same efficiency over 20Hz and 100Hz dataset? Please clarify.*

I have attempted to clarify this characterization of "total data size".

*P5L169-170: Is this the result of the authors' own research? If not, please give a reference.*

I have gone back through my results and observations and changed this statement. I believe that my original intent was to indicate the relationship between friction velocity, vertical flux rate, and ramps times, and mis-stated these observations as being related to stability. I have changed the text and added a short explanation of the rationale, which is not a novel observation (i.e. higher velocity means faster exchange rates).

*P6L174: “again the largest factor in the difference between two methods” is making this phrase unclear. Please change to clarify*

Changed for clarity.

#### **TECHNICAL CORRECTIONS:**

*P1L10: One extra space seems to be typed in between the words “demonstrate” and “that”*

*P1L26: I think “manifests” is necessary instead of “manifest”*

*P2L60: “on” instead of “one”*

*P3L95: “an” instead of “a” before “application”*

*P4L111: Please use different word instead of “conditioning” if possible (i.e. “despiking”)*

*P4L135: Please replace “product of” instead “product by”?*

*P5L165 and 166: are empty.*

*Avoid “@” in Figure titles.*

These technical corrections are appreciated and have been addressed.

*P4L135: Please replace “sense” by “importance in the flux measurement” or similar. Also, replace “generate” by “are”*

The language has been changed for clarity. In particular, the word "sense" was used to indicate vector direction.

*P5L158: Please replace “difference vectors” by “the vectors of the differences”. In addition, change “for each sample lag” by “for the corresponding sample lag”*

This has been changed for clarity as suggested.

General comments from Reviewer #3:

These comments made by this reviewer have helped to make significant improvements in the revised manuscript. In particular, I was unaware of the review of despiking methods by Starkenburg et al. (2016), and this citation has significantly improved the context of this research and has helped narrow the discussion of the intent of my manuscript, and clarify what the novel aspects of the methods described here. Also, the reviewers prompted me to re-read of French et al.'s 2012 publication using the SR method, and the discussion of lag time there has helped improve my discussion of practical aspects of computation. Following the reviewers suggestions, I have re-written the abstract, introduction, and conclusions sections to clarify the intent, context, and relevance of the three computational methods described in the manuscript.

Specific Comments:

*Abstract: Logical presentation in the abstract needs improvement... What is your thesis?*

I acknowledge the lack of clarity and have re-written the abstract to clarify that the intent of the paper is to document and quantify the efficiency of three algorithms, which rely on convolution and algebraic simplifications, and consequently facilitate implementation of the surface renewal method. A secondary finding of the manuscript relates the flux averaging period and computational efficiency. Mention of potential uses for this findings (open source methods, implementing SR on mobile platforms, integration into hardware solutions, etc.) and similar have been moved to the discussion in the conclusions section, rather than being misleadingly referenced in the abstract.

*Not mentioned, but should be, are the SR limitations relative to EC... quantify the benefits: how much less does SR cost?*

I have added discussion relating more context and citations regarding practical implementation of SR to measure flux.

*What fetch benefit does one gain with SR?*

Because it is relevant to the context and motivation for the SR method generally, I have added language to better cite relevant references (Castellví 2012, Göckede et al. 2004; Paw U 2005), and brief discussion regarding the measurement fetch. I have kept this limited as it is not directly relevant to the computational method. In one sense, the method's reduction of measurement fetch is not clearly established in the literature, and following Castellví (2012), it is probably comparable to eddy covariance. However, it is clearly established that SR can give reliable measurements of flux within the roughness sub-layer (Paw U et al. 1995; Katul et al. 1996; Chen et al. 1997; among others more recently). Due to this difference from eddy covariance and gradient methods which require measurements at two heights, the practical source-sink area associated with a measured flux is smaller, and spatial footprints can therefore be resolved at finer scales.

*L43: a major problem here: you have yet to publish your manuscript assessing efficacy of different averaging periods, but as of now we don't know the result and so do not know how important your algorithm refinement is.*

I have expanded the description of the field experiments, and included results that quantify the improved calculation efficiency.

*Regarding de-spiking: authors should read and consider findings in Starkenburg et al. 2016, with attention to their Table 1.*

I am grateful for the reviewer for bringing this paper to my attention as it was published after this manuscript was originally prepared, and significantly adds to the context of the work shown here. While de-spiking is a required QAQC process with micromet data, the convolution method shown here specifically addresses and improves computational efficiency, and shows significantly improvement over the methods shown in Starkenburg's review of published methods. In addition, the convolution approach facilitates a more sophisticated application of the phase space approach, and overcomes a major limitation of that approach through efficient calculation. I have added a brief discussion of my method to the manuscript to demonstrate the benefit of signal processing techniques like convolution.

*Discussion of convolution should mention that the time-series operation includes multiplication of the time-reversed kernel.*

I have added more technical description of convolution.

*Regarding structure function lag estimation, consider findings reported in French et al. discussing lag vs. SR accuracy.*

This citation was also a good suggestion, and I have added a brief discussion of the relevance of lag time to the methods' accuracy, and how this pertains to computational efficiency.

*Root solving is not innovative although I will grant innovation in the diagnostic findings of pathological cubic equations by Edwards and Beaver. It would be useful to compare the speed improvement using Cardano's method vs. numerical Newton-Raphson.*

Although I grant that this would be an interesting comparison, I think that it suffices to state that algebraic solutions are always less computational intensive than any iterative root finding algorithm. Cardano's algebraic solution requires fewer operations than any numerical root finding solution of which I am aware- it is a happy coincidence that the so-called "depressed" cubic polynomial is the one needed to solve the structure function arrangement posed by Van Atta.

*If your aim is open-source you should provide readers with evidence that m script successfully executes on a non-proprietary software platform.*

I agree! In the interest of time, I have not included my partial results of executing these methods in Python, which are ongoing. I rephrased this mention of potential open source implementations to my discussion as a needed action.

*Line 166: here is a critical finding about computation time vs. averaging period; you should find a way to highlight this and not bury it in the text.*

I agree that this is one of the coherent points in the results, and I have reworded to emphasize it.

# Computational Efficiency for the Surface Renewal Method

Jason Kelley\* and Chad Higgins

Dept. of Biological and Ecological Engineering, Oregon State University  
116 Gilmore Hall, Corvallis, OR 97333 USA

\* ~~Corresponding~~ Correspondence to: Jason Kelley (Author, email [kellyja@oregonstate.edu](mailto:kellyja@oregonstate.edu))

## Abstract.

Measuring surface fluxes using the surface renewal (SR) method requires ~~a set of~~ programmatic ~~functions~~ algorithms for tabulation, algebraic calculation, and ~~data~~ quality control. A number of different methods have been published describing automated calibration of SR parameters. ~~Field experiments conducted over a variety of surface conditions used eddy covariance (EC) as a control flux determination for sensible heat flux and evaporation.~~ Because the SR method ~~requires~~ utilizes high frequency (10 Hz+) measurements ~~of a temperature or scalar trace concentration~~, some steps in the flux calculation ~~these tasks~~ are computationally expensive, especially when automating SR to. ~~Some implementations of SR, such as Experimental hypothesis optimization of averaging time, that require perform many iterations analysis of time averaging require many iterations~~ of these calculations. ~~For a study on the minimum time to measure flux, To economize this type of analysis, ss~~ Several new ~~programmatic~~ algorithms ~~methods~~ were written ~~to that~~ to perform the required calculations more efficiently and rapidly, and tested for sensitivity to length of flux averaging period, ability to measure over a large range of lag time scales, and for overall computational efficiency. ~~These algorithms,~~ utilize signal processing techniques and algebraic simplifications that ~~demonstrating~~ simple modifications ~~of that methods~~ dramatically improve computational efficiency. The results here complement efforts by other authors to standardize a robust and accurate computational SR method. Increased ~~Programming techniques~~ speed of computation time ~~such as these can~~ grants flexibility to implementing the SR method, opening new avenues for SR to be ~~applied~~ used in research, for ~~applied~~ monitoring, and in novel field deployments.

## 1 Background Introduction

Originally described by Van Atta (1977), the SR model measures vertical flux that occurs during rapid events which manifest as coherent structures in a turbulent flow. The physical mechanisms are statistically distinct from those described in the ~~Eddy~~ eddy covariance (EC) method, which has been established as ~~a~~ robust and accurate method to measure flux (Baldocechi, 2014) ~~(Baldocechi, 2014).~~, yet ~~†~~ The surface renewal (SR-) method offers several advantages and complements ~~over~~ the use of EC to measure flux. While EC requires ~~a high frequency~~ fast (10 Hz+) measurement of both the vertical wind speed and ~~air~~ temperature to measure the sensible heat flux, the SR method does not explicitly require vertical wind speed, allowing ~~can make this flux~~ to be determined solely ~~determination~~ using ~~from rapid~~ measurements of temperature or other scalar concentrations ~~air temperature measurement only~~. By ~~ing~~ ~~because~~ fewer, lower cost ~~ed~~ sensors are required, ~~cost is d,~~ which ~~expand~~ the SR method theoretically can be

used for ~~direct flux measurement from research applications to more general~~ to develop the SR method for field applications monitoring (Paw U et al., 2005; Spano et al., 2000). Another advantage of SR is ~~over methods such as EC~~ Previous studies have shown ~~isthat~~ the ability to measure flux measurements can be captured very near the surface or near the top of the plant canopy using SR (Katul et al., 1996; Paw U et al., 1992) (Katul et al., 1996; Paw U et al., 1992). By taking measurements very close to the surface, the measurement fetch is effectively reduced and consequently the effective so-called "area from which the flux can be attributed footprint" or contributing area is smaller (Castellví, 2012), yielding a more localized flux estimate.

The SR method estimates turbulent transport rates from fast response measurements of scalar properties such as temperature or trace gas concentration. In the SR conceptual model, rapid changes in scalar concentration are associated with episodic displacement of near-surface air parcels, and the surface condition is renewed from upper air. While in proximity to the surface, the air parcels are gradually enriched or depleted in temperature or scalar concentration by diffusion (Castellví et al., 2002; Paw U et al., 1995). The majority of flux from the surface is attributed to these rapid ejections, which distinguish coherent structures in near surface atmospheric motions (Gao et al., 1989). The duration and amplitude of these rapid fluctuations (visible as ramps in the scalar trace) are used to determine the magnitude and direction of the flux density. ~~As in Taylor's concept of frozen turbulence described by, small scale turbulent transport primarily occurs in during rapid events which manifest as (ramp like coherent structures). Consequently, ipotentially t should anbe possible to measureestimate flux over shorter periods with SR than is typical for EC (15-30 minutes). One possible use of this kind of rapid flux measurement will facilitate would be to map spatially heterogeneous flux using , near surface lightweight sensors mounted on a mobile device. A mobile application of SR could provide missing information for understanding sub-basin scale hydrology, to validate downscaled models, and facilitate efficient water use through heterogeneity precision irrigation systems. Another advantage is by reducing sensor cost, SR can allow more extensive flux measurements than can be accomplished with expensive equipment such as EC systems. By measuring extensively, the spatial heterogeneity of surface fluxes can be explored, and adding to the data collection capacity of sparsely located, regional weather networks. Low cost sensors also facilitate measuring flux in situations where deploying more expensive equipment would be infeasible, such as direct monitoring of farm water use, at unsecured field sites, and in developing regions.~~ standardization of calibration and quality control measures to establish , uniform, and te methods

Because of the short duration of these events, the SR method complements spectral methods to evaluate the flux contributions made over time scales shorter than the typical 15-30 minute averaging time used for EC (Katul et al., 2006; Shapland et al., 2012a, 2012b). Rapid flux measurement will facilitate new applications, such as spatial mapping of flux using vehicle mounted, near-surface sensors, and real-time monitoring systems. Mobile SR implementations and other novel field methods could provide new insights into the complexities of sub-basin scale hydrology, be used to validate downscaled models, and measure the heterogeneity of flux at sub-field scales.

The implementation of ~~specific~~ The use of SR to map heterogeneous surface flux ~~from onto map flux~~ requires a prescribed ~~will first will require establishing knowing a the minimum~~ averaging time period (on the order of minutes) and ramp time duration (on the order of seconds), ~~in which SR can resolve for robust infor which a~~ representative and statistically robust a flux magnitude can be determined ~~can be obtained robustly~~. To implement

SR on a moving vehicle (for instance, to map spatially variable flux), finding a minimum averaging time is desirable to increase the spatial resolution of the resulting map. The averaging time and lag time used in the SR method relate the sensitivity of the scalar measurement to the time scales at which most significant flux occurs (Shapland et al., 2014). To find the minimum ~~required~~ measurement period, ~~several~~ field studies were conducted ~~over in 2014 and -~~  
75 2015 ~~over various types of surface conditions~~. This required a rapid computational method that worked ~~-F, and fluxes were computed~~ over a range of different time averaging periods, and which could implement the various calibration procedures used in the SR method ~~with co-located EC and SR sensors, with several different surface, weather, and crop conditions~~. Initial attempts to calculate flux ~~Initially, the computation followed the~~ methods as described by ~~described in~~ Paw U et al. (2005a) and Snyder et al. (2008) ~~literally~~. However,  
80 implementing these methods as documented ~~but was~~ hampered by ~~slowed by~~ computation time, which constrained ~~for the processing~~ many ~~required~~ iterations required to determine the minimum flux averaging period ~~Paw U et al., 2005 and Snyder et al., 1996~~. However, executing these methods ~~literally as described was constrained by computation found to be inefficient time~~.

As late as the 1990's, limits on computing power, data logger memory, and telemetry limited the implementation of  
85 SR to highly skilled researchers (Katul et al., 1996; Snyder et al., 1996). Open source software and online forums are abundant with methods that utilize advances in computing power, memory availability, and the accessibility of multithreaded processing ~~when calculating over the many iterations that were required for multiple averaging periods~~. ~~More efficient methods were found to utilize the increased computing power that is currently available, and to take advantage of parallel processing approaches~~. These ~~approaches~~ methods reduce computational  
90 overhead, and ~~open possibilities~~ can augment the ~~for~~ SR technique as to allow implementation ~~which can be a constraint when with using~~ low cost ~~data~~ computers and data loggers, or where remote telemetry is required. Three example methods are shown here ~~are shown~~ which ~~clarify~~ streamline specific operational steps in the SR method. The first is a ~~rapid~~ method adapted from signal processing to "despike" noisy data, a quality control technique commonly ~~required~~ used ~~to in~~ processing raw meteorological data. Second is a method to compute  
95 structure functions over multiple time lags rapidly using convolution in two dimensions. Third, an algebraic ~~solution~~ array calculation is used to ~~to find the root~~ find the ~~of a~~ cubic polynomials roots used to ~~, which facilitates the rapid~~ determine the SR ~~ation of SR~~ ramp amplitude ~~as an array calculation~~. By using more efficient algorithms, rapid iterative trials can be conducted to adjust calibration parameters, test hypotheses on the time averaging of flux calculations, and potentially measure SR flux in real-time.

Advantages such as low cost, relatively simple instrumentation, and easier field implementation are all cited as  
100 motivating factors to use the SR method (Paw U et al., 2005), yet work remains to standardize a robust method (French et al., 2012; Suvočarev et al., 2014). Because sensor cost is reduced, SR systems can be implemented to measure flux more extensively than EC, and in situations where EC is impractical. Extensive, site specific SR estimates can augment the utility of sparsely located, permanent weather stations in mapping the heterogeneity of  
105 surface flux. Examples of situations which could benefit from low cost flux measurements include direct crop ET monitoring, experiments at remote field sites, and in developing regions. While SR may expand flux measurement

applications, the method still requires standardized calibration and quality control measures to establish that SR is robust and accurate, and a critical step in developing the method is to reduce computation costs.

## 2 Methods

The example computational approaches-algorithms shown here were adapted to improve or economize existing calculation previously described methods, including despiking of scalar time series data (Højstrup, 1993; Starkenburg et al., 2016)(Højstrup, 1993)(Vickers and Mahrt, 1997), calculation of structure functions (Antonia and Van Atta, 1978)(Antonia and Van Atta, 1978), and Fourier analysis of signals i.e. spectral analysis (Press, 2007; Stull, 1988)(Stull, 1988). In each case, dramatically faster execution times were accomplished using simple programming improvements. Most efficiency gains were due a result of code to "vectorization", which is the conversion of iterative looping algorithms from an iterative loop process into an array calculations. All methods described here were implemented in the Matlab language (The Mathworks Inc., 2016)(The Mathworks Inc., 2016), with including the Statistics, Curve Fitting, and Signal Analysis Toolboxes. Matlab's Profiler (*profile.m*) was used to track the memory demand and speed of time to implement calculations. Trials were conducted on multiple desktop systems; for uniformity, analysis shown here data from only used Test-test runs shown here used in this analysis that were conducted one one system, a in the Windows 10 operating system running on using an Intel Core (TM) i7-3720QM processor operating at 2.60GHz with 16GB RAM. Processor clock speed was verified using the Matlab's Profiler tool at run time, and all processing times reported here include are described as both run time (actual observed execution time) and or as a Total Run Time, which is the sum of CPU time used in for all calculation threads. Example methods are indicated by *function name (in italics)*; and full- Abbreviated, commented code scripts for the example functions is are provided in the supplementary material. Complete code for this analysis can be obtained directly from the corresponding author. Although these methods were prototyped in Matlab, the examples shown are generally useful as solutions to challenges commonly encountered in micrometeorology. Example data was collected during two fvarious field experiments in from 2014 and 2017, using an integrated sonic anemometer and infrared gas analyzer (IRGASON) and fine wire thermocouples (FWTs) which was, and was recorded at 10, Hz and 20, and 100 Hz using a CR1000 datalogger (Campbell Scientific). The data used in verifying the methods is provided through supplementary materials online (DOI: 10.7267/N9X34VDS).

Vector and array calculations are more efficiently executed than iterative methods; Vectorization-vectorization of Matlab code entails removing loops (which are not pre-compiled) and taking advantage of implicit parallel methods in Matlab's pre-compiled library-functions (Altman, 2015)(Altman, 2015). Other significant improvements were enabled through the Fast Fourier Transform by using convolution of number arrays, rather than iterative operations. In the case of determining ramp geometry in the SR method, Cardano's solution for depressed cubic polynomials (published in 1545) reduces a root finding algorithm from an iterative numerical approximation to an algebraic exact algebraic vector vector calculation. While some of the implementation of these programming methods address unique aspects are particular to the Matlab language, the general mathematical concepts are universal. Although

these methods were prototyped in Matlab, the examples shown are generally useful as solutions to challenges commonly encountered in micrometeorology. ~~By reducing computational overhead, these methods expand the possible uses of SR flux measurements to monitor agricultural water use, to measure from lightweight aerial platforms, and in other specialty applications.~~

### **Example 2.1: Despiking of noisy data using convolution**

Despiking is the removal of erroneous or extreme data points from a time series of sampled values. It is a common procedure when measuring environmental parameters, ~~such as wind speed, temperature, and trace gas concentrations~~ especially in challenging conditions or complex environments (Göckede et al., 2004; Starkenburg et al., 2016)(~~Göckede et al., 2004~~). The origin of spikes in a time series ~~can~~ may be or electronic ~~be either~~ or physical (sensor malfunction or actual physical non-errors) ~~or electronic in nature; regardless~~ Regardless of the origin, ~~these~~ spikes can be recorded as ~~either~~ abnormally large or small values, or ~~may be marked by as some~~ a pre-firmware defined error flag. ~~Spikes become problematic which is if they are not readily differentiated when during automatic data automatically importing data~~ (Rebmann et al., 2012)(~~Rebmann et al., 2012~~). Spikes interfere with ~~normal~~ statistical calculations, ~~and requiring~~ require some deliberate and objective method to identify, remove, and interpolate where they exist. For instance, a data logger program may record an error as "9999" or a character string, while Matlab denotes missing values in a numerical array as NaN ("not a number"). Because normally distributed data ~~may contains~~ noise ~~across in~~ a wide range of ~~seals~~ values, robustness of the despiking algorithm is complicated by the requirement to differentiate between "hard spikes" characteristic of automatic flags (such as 9999) and "soft spikes" which are realistically valued, but ~~exceed some objective limit~~ erroneous measurement. ~~This~~ An objective limit for soft spikes is usually defined as appropriate for the signal to noise ratio of any particular data, usually in terms of variance during ~~some a stationary~~ defined windowing period. Clearly distinguishing ~~between these explicit errors and flagged~~ errors can be achieved by a static objective criteria, by a dynamic statistic, or in a separate pre-processing operation. Previous authors have described a variety of methods including ~~use of~~ autocorrelation (Højstrup, 1993)(~~Højstrup, 1993~~)(~~Højstrup, 1993~~) and statistics within a moving window (Vickers and Mahrt, 1997)(~~Vickers and Mahrt, 1997~~). A comprehensive review of despiking methods is presented by Starkenburg et al., (2016), with emphasis on the accuracy and statistical robustness of different computation methods.

Despiking is ~~largely~~ a problem of ~~conditional~~ low pass filtering; consequently this procedure can be treated as an application of signal ~~conditioning processing and which can be therefore~~ performed efficiently using convolution. Mathematically, convolution can be understood as a multiplicative function that combines a data signal with a filter signal. ~~For a discrete signal, the filter is a weighting array, the size of the moving window, which is multiplied (in the Fourier domain) with data inside the a window. In the time domain, the window can be visualized which as it is moving~~ along the data array as it is multiplied. ~~For~~ As examples, a ~~window filter~~ with a weight of 2 at the center of the window, ~~and zeros elsewhere, would amplify the data signal by a factor of two;~~ ~~Ca~~ convolution with a ~~window filter~~ 10 units samples wide, each weighted at 0.1, generates a ~~10 samples~~ running average (~~smoothed~~) of the data ~~over 10 samples~~. The computational efficiency of convolution is a ~~result product~~ of the Fast Fourier

180 Transform (FFT), which allocates memory efficiently by a process known as bit switching. ~~A thorough treatment of bit switching can be found in (Chapters 12 and 13, in of Press (2007) Press, 2007).~~ To demonstrate the increased efficiency of the FFT, two ~~programs-methods~~ were used to despik 8.5 hours of 20 Hz sonic temperature data (609139 samples). ~~A first attempt~~ One method utilized ~~a sliding window in~~ a *for* loop, following the objective criteria described by Vickers and Mahrt (1997). ~~A then improved second program-method~~ (shown in *despike.m*) used convolution to determine a running mean and standard deviation used in the identification of spikes. After 185 multiple runs with different input criteria, the first program average run time was 27 seconds. Using convolution, the second program average run time was 0.2 seconds, decreasing run time by approximately 99%. While this drastic improvement may ~~potentially overemphasize the~~ slow compile times of *for* loops in Matlab (compared to other languages), it nonetheless demonstrates the value of the FFT in ~~speeding calculations with time domain signal conditioningsignals.~~ Faster processing time ~~, and facilitates the abandonment of time saving strategies used in earlier applications of despiking, such as skipping samples or limiting window size~~ more comprehensive, calibrated, and accurate analysis, and can reduce data loss compared to coarser filtering techniques.

190 To ~~measure record~~ test computation time ~~independently of hardware~~ uniformly ~~(processor speed and RAM size),~~ identical 10 Hz data was sub-sampled ~~in to record lengths from~~ of 0.25 to 48 hours, and multiple runs were despiked with each sample set. ~~Raw data was checked for hard error flags which required text to number conversion, but was not otherwise manipulated prior to despiking.~~ Matlab Profiler was used to track the run time for all threads, using the undocumented flag "built-in" to track pre-compiled Matlab functions as well as ~~custom-user~~ functions<sup>1</sup>. The total run time for all threads was tabulated and averaged across sets of each data length (Figure 1). ~~Despiking in despike.m was accomplished with convolution used to calculate the running mean and running standard deviation arrays, which are used as comparators in identifying spikes (see supplementary materials).~~ By using convolution, 200 despiking was two orders of magnitude faster for all lengths of data. To illustrate the effect of Matlab's built-in parallel processes, Figure 2 shows the ratio of actual run time to Total Run Time, indicating that the convolution method relies on computations conducted in parallel for processing increasingly ~~larger longer~~ data records. This benefit is direct accrued from the efficiency of the FFT.

205 With increased computation speed, automatic and accurate despiking can be accomplished, with reduced time cost to determine any necessary calibrate for the procedure. The various methods employed to despik data are variously limited by computational inefficiency (Starkenburg et al., 2016). "Phase space thresholding", originally described by Goring and Nikora (2002) is one such method that Starkenburg noted as being hampered by computational costs, and by a requirement for iterative applications to calibrate despiking parameters. By decreasing the execution time, a similar method was developed that allows rapid and accurate despiking of data, both for the detection of hard and 210 soft spikes. A phase space method allows objective criteria to be calibrated for specific sensor data, and a visual diagnostic phase space diagram that allows for rapid calibration of the despiking criteria (Figure 3). Projecting the signal into a phase space diagram reveals modes related to sensor error, response time, and other factors leading to spikes. Using convolution to determine moving window statistics (such as a moving mean, standard deviation, etc.), objective identification of behaviors characteristic to a particular sensor response. In figure 3, infrared gas analyzer

---

<sup>1</sup> <http://undocumentedmatlab.com/blog/undocumented-profiler-options-part-4>, Accessed January 2017

data (in this case, signal strength) collected at 20Hz for 17 days is projected with one minute moving window statistics. Based on this projection, a cut-off in phase space for spike identification can be assigned, and the subsequent percentage of removed data calculated. In this case, the sensor was repeatedly affected by dust from farm operations (Figure 4), yet only 1.5% of the data was required to be removed as spikes due to the precision of the despiking algorithm. This procedure took less than five seconds of computation.

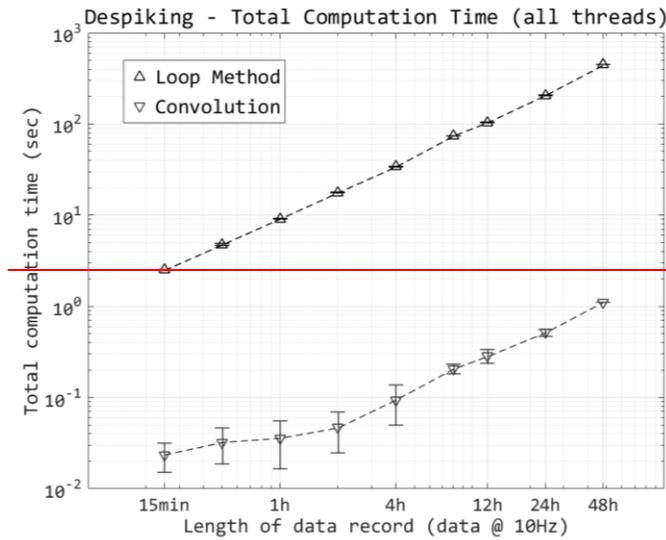


Figure 1: The total computation time is the sum of CPU time spent on all calculation threads. Marker is the mean run time for multiple runs, which varied from 30 runs (15min–4h data) to 10 runs (8, 12, 24 h). 48h calculation is represented by one run only. Error bars represent one standard deviation of all runs.

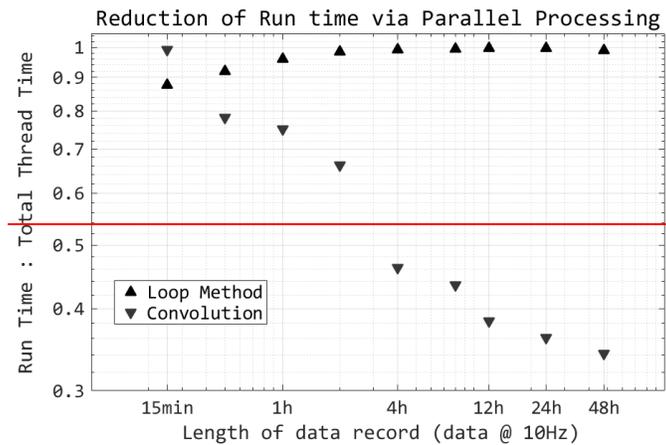


Figure 2: In addition to faster overall run times, calculation of larger data sets remains fast in part due to implicit parallel processing via the Fast Fourier Transform, which can be readily conducted on multiple threads. Parallel processing of noise in the loop method is not apparent.

**Example 2.2: Structure function calculation**

Another computationally-intensive process in SR is the determination of the 2nd, 3rd, and 5th order structure functions. Ramps are an identifiable feature in the measured temperature trace above any natural surface, yet calculating determining the the amplitude of characteristic ramp geometrys from high frequency data requires an

efficient, robust, and preferably automated procedure. There are several methods to determine ramp geometry, including visual detection (Shaw and Gao, 1989)(Shaw and Gao, 1989), low pass filtering (Katul et al., 1996; Paw U et al., 1995)(Katul et al., 1996; Paw U et al., 1995), wavelet analysis (Gao and Li, 1993)(Gao and Li, 1993), and structure functions (Spano et al., 1997)(Spano et al., 1997). Structure functions in particular provide both objective criteria to detect ramps and an efficient method to tabulate ~~ramp geometry~~ statistics of time series data, and use of structure functions has become the predominant method used for SR. The general form for a structure functions is:

$$S^n(r) = \frac{1}{N-r} \sum_1^{i=N-r} [T(i+r) - T(i)]^n \quad \text{Equation-(1)}$$

in which a vector of length  $N-1$  is composed of ~~the~~ differences between sequential (Temperature) samples  $T(i)$ , separated by lag  $r$ . The structure function  $S^n(r)$  of order  $n$  for a given sample ~~time~~ lag  $r$  is obtained by ~~finding differences between each sample  $i$  and sample  $i+r$  (lagged in time by  $r$  samples).~~ ~~The  $n$ th order structure function are defined by the  $n$ th power in the summation~~ raising the difference vector to the  $n$  power, summing the vector and normalizing by  $N-1$ . In a turbulent flow field, ~~The~~ the sampled fluctuations of scalar time series ~~of signal  $\Delta T(i)$  in a turbulent flow can be shown to be~~ a combination of random fluctuations and coherent structures (Van Atta and Park, 1972)(Van Atta and Park, 1972). The random (incoherent) part of the signal is ~~produced~~ a product by of isotropic turbulent processes, and ~~over an a statistically significant~~ ~~adequately large~~ sample this sample should ~~thus~~ have no particular directional sense or orientation (by the isotropic definition). On the other hand, ~~By contrast,~~ coherent structures ~~are by definition~~ generate characteristic anisotropic ~~anisotropies~~ signatures, with periods of gradually ~~ischange~~ punctuated by ~~period and~~ sharp transitions. These sharp transitions occur during "sweeps and ejections" of parcels enriched or depleted in scalar concentration (heat or water trace gas), evidence of transport from an Eulerian perspective. ~~The total variance is composed of additive random and coherent components, and s~~ Structures functions can be used to decompose the ~~measured trace~~ time series fluctuations into ~~random and coherent~~ isotropic and anisotropic components and identify the characteristic ramp amplitude and duration of coherent structures (Van Atta, 1977)(Van Atta, 1977).

Advances in sensor response time and processor speed have revealed an increasingly detailed picture of the coherent ramp structures. In deriving a method to find ramp geometry, Van Atta (1977) calculated structure functions for eight different lags. Two decades later, increased processor power and memory size allowed Snyder et al., (1996) to calculate structure functions on 8 Hz data for lags from 0.25-1.0 seconds, but they were unable to resolve fluxes accurately at some measurement heights and surface roughness conditions. Later it was realized that determining the contributions from "imperfect ramp geometry" would require more thorough examination of ramp durations (Chen et al., 1997a; Paw U et al., 2005)(Chen et al., 1997; Paw U et al., 2005a).

For ~~th~~ this analysis ~~of minimum SR averaging periods~~, data was used from several field experiments. The data records used ~~collected~~ ranged in length ~~over from long measurement periods (8-24 hours)~~ to over two months, with ~~at~~ sampling frequencies of 10Hz, 20Hz, and ~~up to~~ 100Hz (fastest frequency for short duration trials only). Initially, ~~Computation~~ computation of structure functions with the first method (series of nested ~~for~~ loops) for 3 minute periods with lags up to 10 seconds required an average 39 seconds computation ~~time on average with a series of nested loops~~. In contrast, using ~~2-d~~ the convolution method, ~~accomplishes~~ this same calculation was accomplished

in 7.6 seconds, an ~80% reduction in execution time. The function *strfnc.m* (provided in *Supplementary Materials, SI*) also simultaneously time stamps the averaging period, finds the sign of  $S^3(r)$  (used to find flux direction), and indexes the maximized value of  $S^3(r)/r$ , preparing the data for subsequent steps in determining flux. Using 100 Hz FWT data increased the processing time using the convolution method to 38.4 seconds.  ~~demonstrating that the loop based method would be too slow to process 100 Hz data with large data sets or in real time applications, and would require long calculation time when using large continuous data records.~~

For a total of N sample lags, ~~Two~~ two dimensional convolution is ~~accomplished~~ performed with using a filter matrix, which is composed of ~~th~~ with each N column vectors of length N+1: ~~s as~~  $[1 \ -1 \ 0 \ 0 \dots 0], [1 \ 0 \ -1 \ 0 \dots 0], \dots, [1 \ 0 \ 0 \dots -1] [0 \ 0 \dots -1 \dots]$ . ~~Each column in the filter represents~~ ~~a matrix representing~~ a time sample lag increasing distance. When the filter matrix is convolved with time series data, ~~The~~ the resulting column matrix is composed of vectors of the resulting matrix are ~~difference vectors~~ vectors, each representing the difference of the element-wise differences ( $T(i+r) - T(i)$ ) as in Eq. 1; these vectors correspond to each sample lag in the ~~at each sequential lag for~~ ~~(Equation 1)~~ filter. Trials of 10Hz data using Matlab's Profiler showed that calculation efficiency is not accrued directly from convolution, but by changing the order of implementation ~~when multiplying large arrays (for exponentials) and summing.~~ In the looping method, ~~the 2,3, and 5th~~ exponentiation ( $n = 2,3,5$ ) is conducted on the difference vectors for each lag separately. ~~The Exponentiation is implemented~~ accelerated exponentiation in the ~~convolution method via~~ *strfnc.m* is possible by using matrix multiplication of large matrices on the convolved matrix, which and is possible faster ~~once the data is reordered by~~ due to compact memory allocation of the FFT ~~2-d convolution.~~ The resulting efficiency (calculation time for a given data size) ~~is not~~ doesn't dependant on total data size, but is strongly dependent on the length of the averaging period used to ~~calculate flux~~ partition the data (Figure 5) ~~(Figure 3)~~. In other words, the choice of averaging period length is the most significant factor in computation time of the maximized structure functions used to determine ramp geometry. Computation time increases rapidly for periods shorter than 5 minutes. ~~With the ultimate goal of investigating the shortest robust SR measurement period~~ The length of averaging time, ~~this proves to be~~ a critical ~~improvement~~ consideration in ~~iterative SR calculations used to~~ developing a rapid SR measurement method.

In most SR studies to determine flux, a lag time is assigned to the structure function calculation, with only a few authors allowing for a procedure to maximize the ratio  $S^3(r)/r$  (Shapland et al., 2014). Yet lag time has been

identified as a critical parameter in the linear calibration of ramp geometry to calculate flux (French et al., 2012).

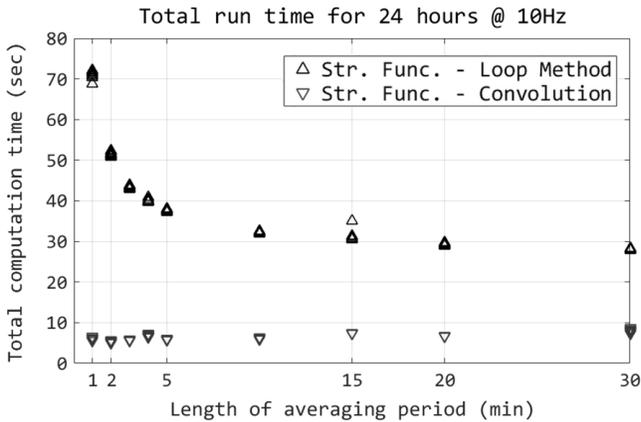
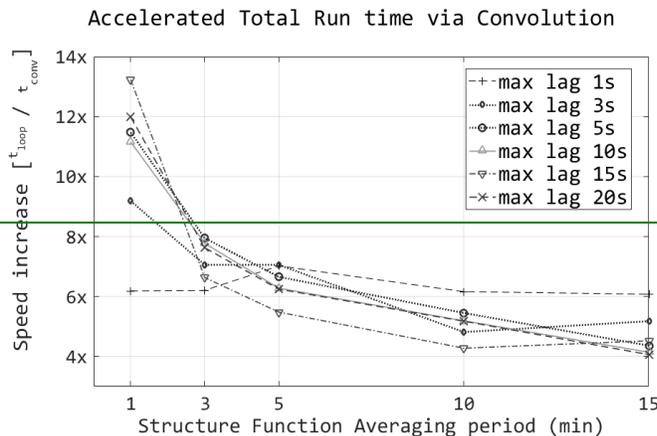


Figure 3: Ten iterations of the structure function calculations using a range of averaging periods.

290 Because the ideal The-SR calculation must identify the lag which maximizes the ratio  $S^3(r)/r$ , the *strfnc.m* procedure calculates also be scripted to evaluate structure functions for a continuous range of lags up to some maximum assigned maximum lag number of lagged samples. Based on repeated trials over a broad range of stability conditions, A-a short maximum lag (3-5 seconds) may be usually adequate under some atmospheric unavailability conditions. Following the model of parcel residence time, this is likely a result of in which buoyancy drives and higher temperature flux magnitude via leading to rapidly transporting coherent structures shorter ramp duration. Under stable conditions, though, longer lags are required to may be important to detect the true maximum of the ratio  $S^3(r)/r$ , indicating that the largest flux contributing time scale contributing to flux increases. To eTo evaluate the sensitivity to the maximum calculated maximum lag, the total run time was structure functions were calculated iteratively, ted with varying the averaging periods and maximum tested lag times (Figure 46). For all Regardless of the length of the assigned range of likely maximum lags, the convolution method outpaced was 300 between 4 and 14 times faster than the loop method by a factor of 4-14, with short averaging periods again the largest factor in the difference in total run time between the two methods. Using the 2-d convolution, automated selection of a lag in a continuous time series is feasible.



305 ~~Figure 4: The performance gains using convolution are more significant for short averaging periods, regardless of maximum lag used in calculating structure functions.~~

### **Example Method 2.3: Cardano's method for depressed cubic polynomials**

For the idealized SR method, the ~~calculated~~ structure functions are ~~retained~~ (for each averaging period) ~~at from for some the~~ lag which maximizes the ratio  $S^3(r)/r$  - this lag is associated with the theoretical maximum contributing scale of flux. The resulting values are used as coefficients in a cubic polynomial, the root of which is the ramp amplitude (A) used to calculate flux:

$$A^3 + \left(10S^2(r) - \frac{S^5(r)}{S^3(r)}\right)A + 10S^3(r) = 0 \quad \text{Equation-(2)}$$

The magnitude of the real root (of 3 possible roots) ~~is regarded as the corresponding solution for~~ the characteristic ramp amplitude of the scalar trace (Spano et al., 1997) ~~(Spano et al., 1997)~~. The Matlab root finding algorithm computes eigenvalues of a companion matrix to approximate the solution to a  $n^{\text{th}}$  order polynomial, regarding the input function as a vector with  $n+1$  elements (*roots.m* documentation<sup>2</sup>). Consequently, this function cannot be executed directly on an array. On the other hand, ~~The an~~ algebraic solution method can be applied to vectors. An appropriate method for this type of cubic polynomial was ~~proposed~~ found by Gerolamo Cardano in the 1545 *Arts Magna*. Cardano's solution for "depressed" cubics (with no squared term), ~~on the other hand, can be applied to vectors. Cardano's solution~~ is found by substituting A with  $(m^{1/3} + n^{1/3})$  into the abbreviated equation  $A^3 + pA + q = 0$ . Expanding terms and using the quadratic equation yields an exact ~~(complex)~~ solution:

$$A = \left(-\frac{q}{2} + \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}\right)^{\frac{1}{3}} + \left(-\frac{q}{2} - \sqrt{\left(\frac{q}{2}\right)^2 + \left(\frac{p}{3}\right)^3}\right)^{\frac{1}{3}} \quad \text{Equation-(3)}$$

where **p** and **q** are ~~the~~ coefficients ~~of the in the~~ depressed cubic and derived from the structure functions (Edwards and Beaver, 2015) ~~(Edwards and Beaver, 2015)~~. The function *cardanos.m* was adapted from a function by Bruno Luong<sup>3</sup>, in a reduced form for the real valued cases used to implement the SR method. The function output was verified against the Matlab function *roots.m* for polynomials with both positive and negative real valued inputs (imaginary inputs are applicable to ramp parameters). Solution for the real roots in this manner expedites ~~rapid SR flux calculation~~ determination of ~~ons~~ flux magnitude and ~~and facilitates the determination of flux~~ direction. ~~The The decreased time of implementation is trivial in this example; rather,~~ algebraic root finding method ~~on arrays dramatically~~ simplifies and speeds iterative application of the SR method by operating directly on arrays. Solving for the roots of this function yields a single, predominant ramp amplitude from a given temperature trace (with units of °C or K). In addition to ramp amplitude, the time scale or ramp duration must also be determined. (Van Atta, 1977) suggested that ramp time  $\tau$  should be related linearly to amplitude **A**, and proposed:

<sup>2</sup> <http://www.mathworks.com/help/matlab/ref/roots.html>, Accessed 9AUG2016

<sup>3</sup> [https://www.mathworks.com/matlabcentral/newsreader/view\\_thread/165013](https://www.mathworks.com/matlabcentral/newsreader/view_thread/165013), Accessed 10MAY2016

$$\tau = \frac{-A^3 r}{S^3(r)} \quad (4)$$

In practice, determination of ramp time  $\tau$  from  $A$  using this equation requires an empirical calibration; this calibration has been shown to be related to surface conditions and instrumentation (Chen et al., 1997b; Shapland et al., 2014). Ongoing work using replicate measurements at multiple heights (Castellvi, 2004) and frequency response calibration (Shapland et al., 2014; Suvočarev et al., 2014) have begun to resolve the causes of variability in this parameter. In this study, it was found that the ratio in equation 4 remains essentially constant for a given surface roughness condition, allowing determination of  $\tau$  algebraically. Automated computation of equation 2 using the exact solution facilitates rapid evaluation of the ramp geometry, and determining flux magnitude from ramp geometry is a relatively simple matter of linear scaling when calibrating to a control measure such as eddy covariance.

### 3 Conclusions

As with other methods for measuring flux from the surface flow parameters such as flux and turbulent, analytic solutions do not always translate easily into straightforward numerical computation, especially when working with large data records or when calculating in real-time of high frequency data. In applied research, despite advances in data storage, memory size, and processor speed, the basic custom programming methods used algorithms for analysis are often often developed by individual researchers, requiring special who often do not have training in programming, significant time investment, or and the motivation to develop use sophisticated programming method techniques that fully utilize available memory and processing power. Efforts to standardize the eddy covariance method (Aubinet et al., 2012; Baldocchi, 2014) (Aubinet et al., 2012; Baldocchi, 2014) and data quality control (Allen et al., 2011; Foken et al., 2012) (Allen et al., 2011; Foken et al., 2012) have not yet extended been similarly applied to the SR method, although substantial work has been made to validate calibration and unify SR methods and field methods (Castellví, 2012; Chen et al., 1997b; French et al., 2012; Suvočarev et al., 2014) (Castellví, 2012; Paw U et al., 2005b; Shapland et al., 2014). By appropriating methods common to other disciplines such as in signal processing, and by sharing open source tools using online forums such as stackexchange.com<sup>4</sup>, more sophisticated approaches can be implemented more broadly. One goal of these open source efforts should be to In particular, reducing the computational overhead of calculating flux enables, so as to facilitate the broadest implementation and robust verification of the SR method in applied contexts. Rapid algorithms allow for automated assignment of lag time, rather than fixed assignment, and allow flux determinations while varying the length of flux averaging periods. These procedures allow for comprehensive analysis of both the physical time scales of surfaces flux, and the sensor response and uncertainty associated with the SR derived flux. Calibration of despiking criteria can be implemented quickly at low computational cost. In summary, efficient methods for computing SR flux allow implementation in novel deployments such as low cost, continuous

---

<sup>4</sup> need this website

365 monitoring, and on moving platforms. Future work remains to transfer efficient methods from the Matlab  
development platform to open source implementations, and enable hardware to perform these techniques directly for  
real-time applications. Reducing the cost and power requirement of the required data loggers, computers, and  
telemetry, will facilitate the extensive deployment of SR sensors to aid in describing the heterogeneity of flux across  
the landscape.

#### 4 Data Availability

370 All data used in this analysis and scripts implementing the algorithms described above are available online at  
<http://hdl.handle.net/1957/60599>. This supplementary materials is also listed with DOI: 10.7267/N9X34VDS  
Abbreviated scripts for the three example methods may be found in the supplemental materials. Requests for phase-  
space despiking methods can be directed to the corresponding author.

#### 375 References

Allen, R., Pereira, L. S., Howell, T. A. and Jensen, M. E.: Evapotranspiration information reporting: I. Factors  
governing measurement accuracy, *Agricultural Water Management*, 98(6), 899–920, 2011.

Altman, Y.: *Accelerating Matlab Performance*, CRC Press., 2015.

380 Antonia, R. A. and Van Atta, C. W.: Structure functions of temperature fluctuations in turbulent shear flows, *Journal  
of Fluid Mechanics*, 84(03), 561–580, 1978.

Aubinet, M., Vesala, T. and Papale, D.: *Eddy covariance: a practical guide to measurement and data analysis*,  
Springer Science & Business Media., 2012.

Baldocchi, D.: Measuring fluxes of trace gases and energy between ecosystems and the atmosphere—the state and  
future of the eddy covariance method, *Global change biology*, 20(12), 3600–3609, 2014.

385 Castellvi, F.: Combining surface renewal analysis and similarity theory: a new approach for estimating sensible heat  
flux, *Water resources research*, 40(5), 2004.

Castellví, F.: Fetch requirements using surface renewal analysis for estimating scalar surface fluxes from  
measurements in the inertial sublayer, *Agricultural and Forest Meteorology*, 152, 233–239,  
doi:10.1016/j.agrformet.2011.10.004, 2012.

390 Castellví, F., Perez, P. J. and Ibañez, M.: A method based on high-frequency temperature measurements to estimate  
the sensible heat flux avoiding the height dependence, *Water Resources Research*, 38(6), 20-1-20-9, 2002.

Chen, W., Novak, M., Black, T. A. and Lee, X.: Coherent eddies and temperature structure functions for three  
contrasting surfaces. Part I: Ramp model with finite microfront time, *Boundary-Layer Meteorology*, 84(1), 99–124,  
doi:10.1023/A:1000338817250, 1997a.

395 Chen, W., Novak, M., Black, T. A. and Lee, X.: Coherent eddies and temperature structure functions for three  
contrasting surfaces. Part II: Renewal model for sensible heat flux, *Boundary-Layer Meteorology*, 84(1), 125–147,  
doi:10.1023/A:1000342918158, 1997b.

Edwards, A. C. and Beaver, J. M.: *Investigating Cardano’s Irreducible Case*, 2015 NCUR, 2015.

400 Foken, T., Leuning, R., Oncley, S. R., Mauder, M. and Aubinet, M.: Corrections and data quality control, in *Eddy  
Covariance*, pp. 85–131, Springer., 2012.

- French, A. N., Alfieri, J. G., Kustas, W. P., Prueger, J. H., Hipps, L. E., Chávez, J. L., Evett, S. R., Howell, T. A., Gowda, P. H., Hunsaker, D. J. and Thorp, K. R.: Estimation of surface energy fluxes using surface renewal and flux variance techniques over an advective irrigated agricultural site, *Advances in Water Resources*, 50, 91–105, doi:10.1016/j.advwatres.2012.07.007, 2012.
- 405 Gao, W. and Li, B. L.: Wavelet analysis of coherent structures at the atmosphere-forest interface, *Journal of Applied Meteorology*, 32(11), 1717–1725, 1993.
- Gao, W., Shaw, R. H. and Paw U, K. T.: Observation of organized structure in turbulent flow within and above a forest canopy, *Boundary-Layer Meteorology*, 47(1), 349–377, 1989.
- 410 Göckede, M., Rebmann, C. and Foken, T.: A combination of quality assessment tools for eddy covariance measurements with footprint modelling for the characterisation of complex sites, *Agricultural and Forest Meteorology*, 127(3), 175–188, 2004.
- Goring, D. G. and Nikora, V. I.: Despiking Acoustic Doppler Velocimeter Data, *Journal of Hydraulic Engineering*, 128(1), 117–126, doi:10.1061/(ASCE)0733-9429(2002)128:1(117), 2002.
- Højstrup, J.: A statistical data screening procedure, *Measurement Science and Technology*, 4(2), 153, 1993.
- 415 Katul, G., Hsieh, C.-I., Oren, R., Ellsworth, D. and Phillips, N.: Latent and sensible heat flux predictions from a uniform pine forest using surface renewal and flux variance methods, *Boundary-Layer Meteorology*, 80(3), 249–282, 1996.
- Katul, G., Porporato, A., Cava, D. and Siqueira, M.: An analysis of intermittency, scaling, and surface renewal in atmospheric surface layer turbulence, *Physica D: Nonlinear Phenomena*, 215(2), 117–126, 2006.
- 420 Paw U, K. T., Brunet, Y., Collineau, S., Shaw, R. H., Maitani, T., Qiu, J. and Hipps, L.: On coherent structures in turbulence above and within agricultural plant canopies, *Agricultural and Forest Meteorology*, 61(1–2), 55–68, doi:10.1016/0168-1923(92)90025-Y, 1992.
- Paw U, K. T., Qiu, J., Su, H.-B., Watanabe, T. and Brunet, Y.: Surface renewal analysis: a new method to obtain scalar fluxes, *Agricultural and Forest Meteorology*, 74(1–2), 119–137, doi:10.1016/0168-1923(94)02182-J, 1995.
- 425 Paw U, K. T., Snyder, R. L., Spano, D. and Su, H.-B.: Surface Renewal Estimates of Scalar Exchange, in *Micrometeorology in Agricultural Systems*, pp. 455–483, American Society of Agronomy, Crop Science Society of America, and Soil Science Society of America, Madison, WI. [online] Available from: <http://dx.doi.org/10.2134/agronmonogr47.c20>, 2005.
- 430 Press, W. H., Ed.: *Numerical recipes: the art of scientific computing*, 3rd ed., Cambridge University Press, Cambridge, UK ; New York., 2007.
- Rebmann, C., Kolle, O., Heinesch, B., Queck, R., Ibrom, A. and Aubinet, M.: Data acquisition and flux calculations, in *Eddy Covariance*, pp. 59–83, Springer., 2012.
- 435 Shapland, T. M., McElrone, A. J., Snyder, R. L. and Paw U, K. T.: Structure Function Analysis of Two-Scale Scalar Ramps. Part I: Theory and Modelling, *Boundary-Layer Meteorol*, 145(1), 5–25, doi:10.1007/s10546-012-9742-5, 2012a.
- Shapland, T. M., McElrone, A. J., Snyder, R. L. and Paw U, K. T.: Structure Function Analysis of Two-Scale Scalar Ramps. Part II: Ramp Characteristics and Surface Renewal Flux Estimation, *Boundary-Layer Meteorol*, 145(1), 27–44, doi:10.1007/s10546-012-9740-7, 2012b.
- 440 Shapland, T. M., Snyder, R. L., Paw U, K. T. and McElrone, A. J.: Thermocouple frequency response compensation leads to convergence of the surface renewal alpha calibration, *Agricultural and Forest Meteorology*, 189–190, 36–47, doi:10.1016/j.agrformet.2014.01.008, 2014.
- Shaw, R. H. and Gao, W.: Detection of temperature ramps and flow structures at a deciduous forest site, *Agricultural and forest meteorology*, 47(2), 123–138, 1989.
- 445 Snyder, R. L., Spano, D. and Pawu, K. T.: Surface renewal analysis for sensible and latent heat flux density, *Boundary-Layer Meteorology*, 77(3–4), 249–266, 1996.

- Snyder, R. L., Spano, D., Duce, P., Paw U, K. T. and Rivera, M.: Surface renewal estimation of pasture evapotranspiration, *Journal of Irrigation and Drainage Engineering*, 134(6), 716–721, 2008.
- Spano, D., Snyder, R. L., Duce, P. and Paw U, K. T.: Surface renewal analysis for sensible heat flux density using structure functions, *Agricultural and Forest Meteorology*, 86(3), 259–271, 1997.
- 450 Spano, D., Snyder, R. L. and Duce, P.: Estimating sensible and latent heat flux densities from grapevine canopies using surface renewal, *Agricultural and Forest Meteorology*, 104(3), 171–183, 2000.
- Starkenburg, D., Metzger, S., Fochesatto, G. J., Alfieri, J. G., Gens, R., Prakash, A. and Cristóbal, J.: Assessment of Despiking Methods for Turbulence Data in Micrometeorology, *Journal of Atmospheric and Oceanic Technology*, 33(9), 2001–2013, doi:10.1175/JTECH-D-15-0154.1, 2016.
- 455 Stull, R. B.: *An introduction to boundary layer meteorology*, Springer., 1988.
- Suvočarev, K., Shapland, T. M., Snyder, R. L. and Martínez-Cob, A.: Surface renewal performance to independently estimate sensible and latent heat fluxes in heterogeneous crop surfaces, *Journal of Hydrology*, 509, 83–93, doi:10.1016/j.jhydrol.2013.11.025, 2014.
- The Mathworks Inc.: *Matlab R2016b*, The MathWorks Inc., Natick, MA., 2016.
- 460 Van Atta, C. W.: Effect of coherent structures on structure functions of temperature in the atmospheric boundary layer, *Archiwum Mechaniki Stosowanej*, 29(1), 161–171, 1977.
- Van Atta, C. W. and Park, J.: Statistical self-similarity and inertial subrange turbulence, in *Statistical Models and Turbulence*, pp. 402–426, Springer., 1972.
- 465 Vickers, D. and Mahrt, L.: Quality control and flux sampling problems for tower and aircraft data, *Journal of Atmospheric and Oceanic Technology*, 14(3), 512–526, 1997.

## Figures

Allen, R., Pereira, L.S., Howell, T.A., Jensen, M.E., 2011. Evapotranspiration information reporting: I. Factors governing measurement accuracy. *Agricultural Water Management* 98, 899–920.

Altman, Y., 2015. *Accelerating Matlab Performance*. CRC Press.

Antonia, R.A., Van Atta, C.W., 1978. Structure functions of temperature fluctuations in turbulent shear flows. *Journal of Fluid Mechanics* 84, 561–580.

Aubinet, M., Vesala, T., Papale, D., 2012. *Eddy covariance: a practical guide to measurement and data analysis*. Springer Science & Business Media.

Baldocchi, D., 2014. Measuring fluxes of trace gases and energy between ecosystems and the atmosphere—the state and future of the eddy covariance method. *Global change biology* 20, 3600–3609.

Castellví, F., 2012. Fetch requirements using surface renewal analysis for estimating scalar surface fluxes from measurements in the inertial sublayer. *Agricultural and Forest Meteorology* 152, 233–239. doi:10.1016/j.agrformet.2011.10.004

Chen, W., Novak, M., Black, T.A., Lee, X., 1997. Coherent eddies and temperature structure functions for three contrasting surfaces. Part I: Ramp model with finite microfront time. *Boundary Layer Meteorology* 84, 99–124. doi:10.1023/A:1000338817250

Edwards, A.C., Beaver, J.M., 2015. *Investigating Cardano's Irreducible Case*. 2015 NCUR.

Foken, T., Leuning, R., Oneley, S.R., Mauder, M., Aubinet, M., 2012. Corrections and data quality control, in: *Eddy Covariance*. Springer, pp. 85–131.

Gao, W., Li, B.L., 1993. Wavelet analysis of coherent structures at the atmosphere-forest interface. *Journal of Applied Meteorology* 32, 1717–1725.

Göckede, M., Rebmann, C., Foken, T., 2004. A combination of quality assessment tools for eddy covariance measurements with footprint modelling for the characterisation of complex sites. *Agricultural and Forest Meteorology* 127, 175–188.

Højstrup, J., 1993. A statistical data screening procedure. *Measurement Science and Technology* 4, 153.

Katul, G., Hsieh, C.-I., Oren, R., Ellsworth, D., Phillips, N., 1996. Latent and sensible heat flux predictions from a uniform pine forest using surface renewal and flux variance methods. *Boundary Layer Meteorology* 80, 249–282.

Paw U, K.T., Brunet, Y., Collineau, S., Shaw, R.H., Maitani, T., Qiu, J., Hipps, L., 1992. On coherent structures in turbulence above and within agricultural plant canopies. *Agricultural and Forest Meteorology* 61, 55–68. doi:10.1016/0168-1923(92)90025-Y

Paw U, K.T., Qiu, J., Su, H. B., Watanabe, T., Brunet, Y., 1995. Surface renewal analysis: a new method to obtain scalar fluxes. *Agricultural and Forest Meteorology* 74, 119–137. doi:10.1016/0168-1923(94)02182-J

Paw U, K.T., Snyder, R.L., Spano, D., Su, H. B., 2005a. Surface Renewal Estimates of Scalar Exchange, in: *Micrometeorology in Agricultural Systems*, Agronomy Monograph. American Society of Agronomy, Crop Science Society of America, and Soil Science Society of America, Madison, WI, pp. 455–483.

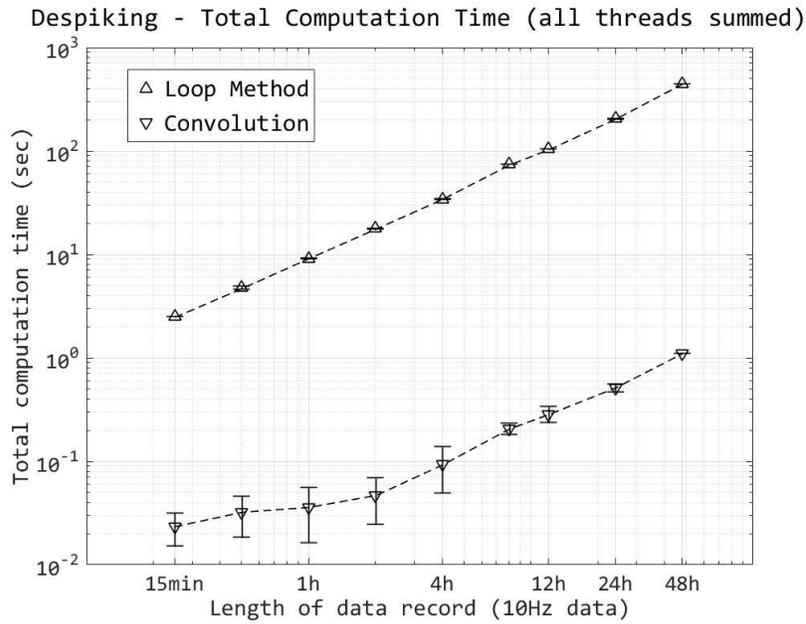
Paw U, K.T., Snyder, R.L., Spano, D., Su, H. B., 2005b. Surface Renewal Estimates of Scalar Exchange, in: *Micrometeorology in Agricultural Systems*. American Society of Agronomy, Crop Science Society of America, and Soil Science Society of America, Madison, WI, pp. 455–483.

Press, W.H. (Ed.), 2007. *Numerical recipes: the art of scientific computing*, 3rd ed. ed. Cambridge University Press, Cambridge, UK ; New York.

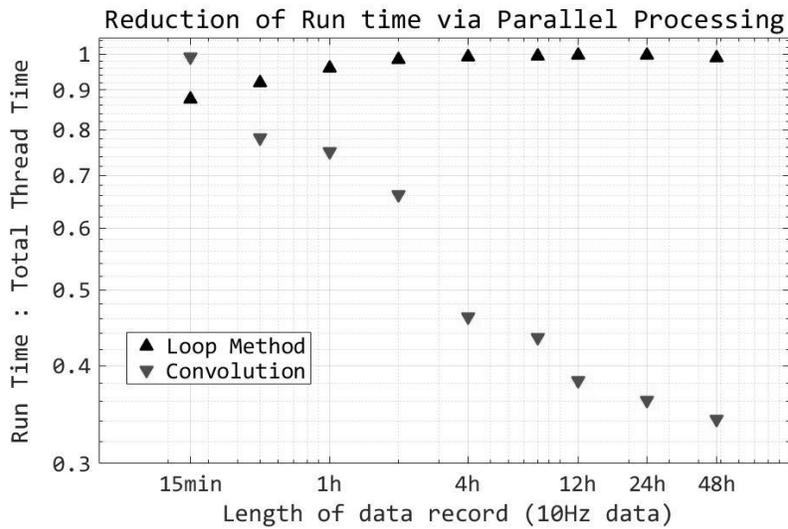
- Rebmann, C., Kolle, O., Heinesch, B., Queck, R., Ibrom, A., Aubinet, M., 2012. Data acquisition and flux calculations, in: *Eddy Covariance*. Springer, pp. 59–83.
- Shapland, T.M., Snyder, R.L., Paw U, K.T., McElrone, A.J., 2014. Thermocouple frequency response compensation leads to convergence of the surface renewal alpha calibration. *Agricultural and Forest Meteorology* 189–190, 36–47. doi:10.1016/j.agrformet.2014.01.008
- Shaw, R.H., Gao, W., 1989. Detection of temperature ramps and flow structures at a deciduous forest site. *Agricultural and forest meteorology* 47, 123–138.
- Snyder, R.L., Spano, D., Duce, P., Paw U, K.T., Rivera, M., 2008. Surface renewal estimation of pasture evapotranspiration. *Journal of Irrigation and Drainage Engineering* 134, 716–721.
- Snyder, R.L., Spano, D., Pawu, K.T., 1996. Surface renewal analysis for sensible and latent heat flux density. *Boundary-Layer Meteorology* 77, 249–266.
- Spano, D., Snyder, R.L., Duce, P., 2000. Estimating sensible and latent heat flux densities from grapevine canopies using surface renewal. *Agricultural and Forest Meteorology* 104, 171–183.
- Spano, D., Snyder, R.L., Duce, P., Paw U, K.T., 1997. Surface renewal analysis for sensible heat flux density using structure functions. *Agricultural and Forest Meteorology* 86, 259–271.
- Stull, R.B., 1988. *An introduction to boundary layer meteorology*. Springer.
- The Mathworks Inc., 2016. *Matlab R2016b*. The MathWorks Inc., Natick, MA.
- Van Atta, C.W., 1977a. Effect of coherent structures on structure functions of temperature in the atmospheric boundary layer. *Archiwum Mechaniki Stosowanej* 29, 161–171.
- Van Atta, C.W., 1977b. Effect of coherent structures on structure functions of temperature in the atmospheric boundary layer. *Archiwum Mechaniki Stosowanej* 29, 161–171.
- Van Atta, C.W., Park, J., 1972. Statistical self-similarity and inertial subrange turbulence, in: *Statistical Models and Turbulence*. Springer, pp. 402–426.
- Vickers, D., Mahrt, L., 1997. Quality control and flux sampling problems for tower and aircraft data. *Journal of Atmospheric and Oceanic Technology* 14, 512–526.
- Allen, R., Pereira, L.S., Howell, T.A., Jensen, M.E., 2011. Evapotranspiration information reporting: I. Factors governing measurement accuracy. *Agricultural Water Management* 98, 899–920.
- Altman, Y., 2015. *Accelerating Matlab Performance*. CRC Press.
- Antonia, R.A., Van Atta, C.W., 1978. Structure functions of temperature fluctuations in turbulent shear flows. *Journal of Fluid Mechanics* 84, 561–580.
- Aubinet, M., Vesala, T., Papale, D., 2012. *Eddy covariance: a practical guide to measurement and data analysis*. Springer Science & Business Media.
- Baldocchi, D., 2014. Measuring fluxes of trace gases and energy between ecosystems and the atmosphere—the state and future of the eddy covariance method. *Global change biology* 20, 3600–3609.
- Castellví, F., 2012. Fetch requirements using surface renewal analysis for estimating scalar surface fluxes from measurements in the inertial sublayer. *Agricultural and Forest Meteorology* 152, 233–239. doi:10.1016/j.agrformet.2011.10.004
- Chen, W., Novak, M., Black, T.A., Lee, X., 1997. Coherent eddies and temperature structure functions for three contrasting surfaces. Part I: Ramp model with finite microfront time. *Boundary-Layer Meteorology* 84, 99–124. doi:10.1023/A:1000338817250
- Edwards, A.C., Beaver, J.M., 2015. *Investigating Cardano's Irreducible Case*. 2015 NCUR.

- Foken, T., Leuning, R., Oneley, S.R., Mauder, M., Aubinet, M., 2012. Corrections and data quality control, in: *Eddy Covariance*. Springer, pp. 85–131.
- Gao, W., Li, B.L., 1993. Wavelet analysis of coherent structures at the atmosphere-forest interface. *Journal of Applied Meteorology* 32, 1717–1725.
- Göckede, M., Rebmann, C., Foken, T., 2004. A combination of quality assessment tools for eddy covariance measurements with footprint modelling for the characterisation of complex sites. *Agricultural and Forest Meteorology* 127, 175–188.
- Hojstrup, J., 1993. A statistical data screening procedure. *Measurement Science and Technology* 4, 153.
- Katul, G., Hsieh, C.-I., Oren, R., Ellsworth, D., Phillips, N., 1996. Latent and sensible heat flux predictions from a uniform pine forest using surface renewal and flux variance methods. *Boundary-Layer Meteorology* 80, 249–282.
- Paw U, K.T., Brunet, Y., Collineau, S., Shaw, R.H., Maitani, T., Qiu, J., Hipps, L., 1992. On coherent structures in turbulence above and within agricultural plant canopies. *Agricultural and Forest Meteorology* 61, 55–68. doi:10.1016/0168-1923(92)90025-Y
- Paw U, K.T., Qiu, J., Su, H.-B., Watanabe, T., Brunet, Y., 1995. Surface renewal analysis: a new method to obtain scalar fluxes. *Agricultural and Forest Meteorology* 74, 119–137. doi:10.1016/0168-1923(94)02182-J
- Paw U, K.T., Snyder, R.L., Spano, D., Su, H.-B., 2005a. Surface Renewal Estimates of Scalar Exchange, in: *Micrometeorology in Agricultural Systems*, Agronomy Monograph. American Society of Agronomy, Crop Science Society of America, and Soil Science Society of America, Madison, WI, pp. 455–483.
- Paw U, K.T., Snyder, R.L., Spano, D., Su, H.-B., 2005b. Surface Renewal Estimates of Scalar Exchange, in: *Micrometeorology in Agricultural Systems*. American Society of Agronomy, Crop Science Society of America, and Soil Science Society of America, Madison, WI, pp. 455–483.
- Press, W.H. (Ed.), 2007. *Numerical recipes: the art of scientific computing*, 3rd ed. ed. Cambridge University Press, Cambridge, UK ; New York.
- Rebmann, C., Kolke, O., Heinesch, B., Queck, R., Ibrom, A., Aubinet, M., 2012. Data acquisition and flux calculations, in: *Eddy Covariance*. Springer, pp. 59–83.
- Shapland, T.M., Snyder, R.L., Paw U, K.T., McElrone, A.J., 2014. Thermocouple frequency response compensation leads to convergence of the surface renewal alpha calibration. *Agricultural and Forest Meteorology* 189–190, 36–47. doi:10.1016/j.agrformet.2014.01.008
- Shaw, R.H., Gao, W., 1989. Detection of temperature ramps and flow structures at a deciduous forest site. *Agricultural and forest meteorology* 47, 123–138.
- Snyder, R.L., Spano, D., Pawu, K.T., 1996. Surface renewal analysis for sensible and latent heat flux density. *Boundary-Layer Meteorology* 77, 249–266.
- Spano, D., Snyder, R.L., Duce, P., Paw U, K.T., 1997. Surface renewal analysis for sensible heat flux density using structure functions. *Agricultural and Forest Meteorology* 86, 259–271.
- Stull, R.B., 1988. *An introduction to boundary layer meteorology*. Springer.
- The Mathworks Inc., 2016. *Matlab R2016b*. The MathWorks Inc., Natick, MA.
- Van Atta, C.W., 1977. Effect of coherent structures on structure functions of temperature in the atmospheric boundary layer. *Archiwum Mechaniki Stosowanej* 29, 161–171.
- Van Atta, C.W., Park, J., 1972. Statistical self-similarity and inertial subrange turbulence, in: *Statistical Models and Turbulence*. Springer, pp. 402–426.
- Vickers, D., Mahrt, L., 1997. Quality control and flux sampling problems for tower and aircraft data. *Journal of Atmospheric and Oceanic Technology* 14, 512–526.

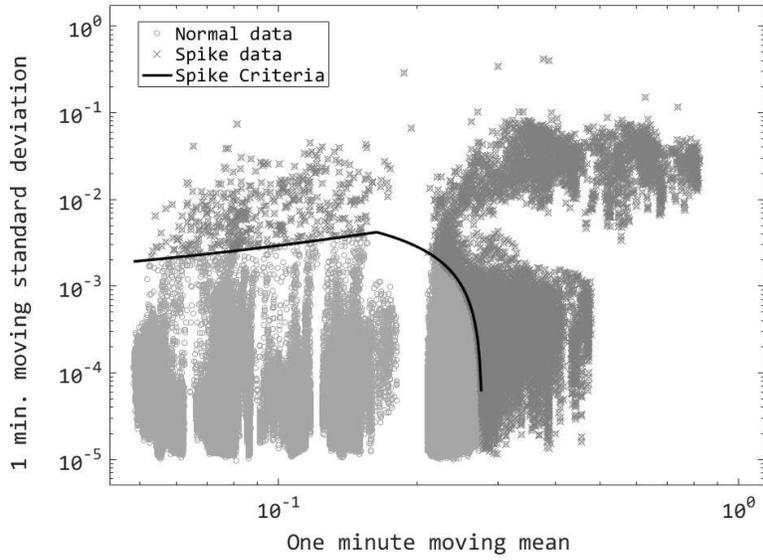




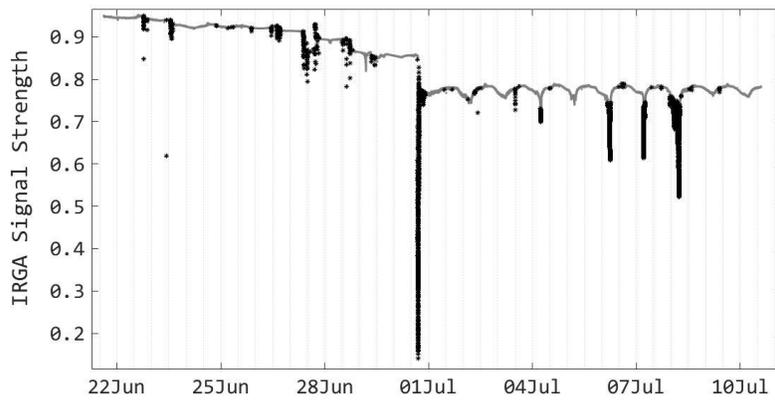
**Figure 1: The total computation time is the sum of CPU time spent on all calculation threads. Marker is the mean run time for multiple runs, which varied from 30 runs (15min - 4h data) to 10 runs (8, 12, 24 h). 48h calculation is represented by one run only. Error bars represent one standard deviation of all runs.**



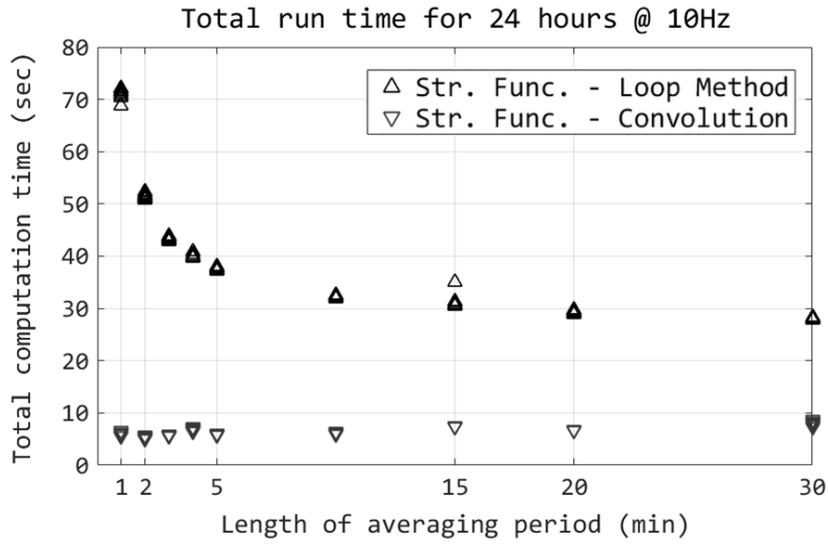
**Figure 2: Fast calculation of larger data sets is due to implicit parallel processing via the FFT, which is readily performed by multiple simultaneous threads. The efficiency of parallel processing is shown by a lower ratio of Run Time to Total Thread Time.**



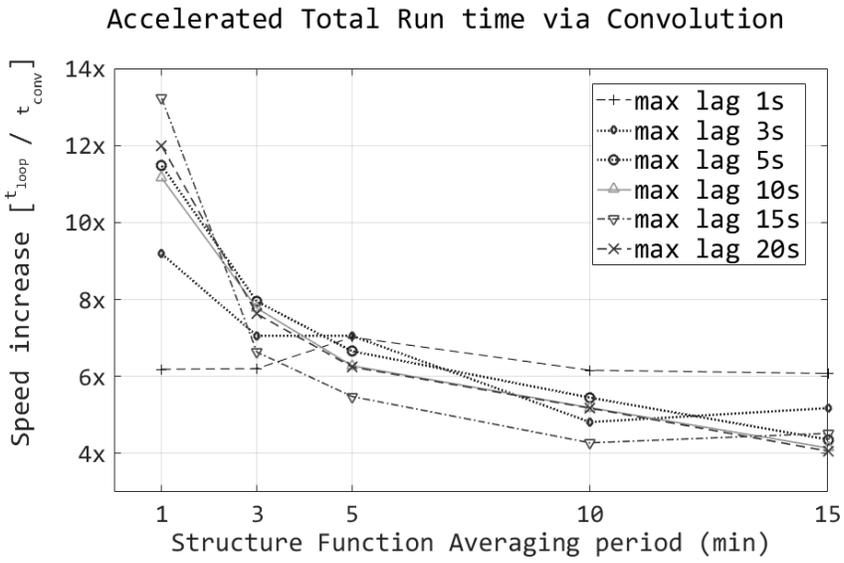
**Figure 3: Phase space diagram showing moving window statistics of IRGA signal (17 days of 20 Hz data)**



**Figure 4: Time series showing data removed as spikes (bolded) from phase space criteria in Figure 3.**



**Figure 5:** Ten iterations of the structure function calculations using a range of averaging periods.



**Figure 6:** The performance gains using convolution are more significant for short averaging periods, regardless of maximum lag used in calculating structure functions.

## Supplementary Materials

### S.1 Despiking method using convolution (*despike.m*)

```
1 function [data_ds, ns, index] = despike(data, nw, sig, buffer, varargin)
2
3 % DESPIKE filters out spikes from a data vector using a Gaussian convolution.
4 %   INPUTS: data: nx1 vector
5 %           nw: sample size of "sliding window" used for convolution
6 %           sig: # of standard deviations considered significant (& removed)
7 %           buff: number of adjacent samples to remove
8 %           varargin{1}: 'interp' option interpolates over nans (cubic)
9 %           varargin{2}: timestamps for data
10 %           varargin{3}: (optional) passes interpolation method ('cubic', etc.)
11 %                       w/o 3rd varargin, reverts to default 'linear'
12 %   OUTPUTS:           data_ds: despiked data
13 %           ns: number of (removed) spikes
14 %           index: logical vector with TRUE = spike
15 %   REQUIRES: setnan.m (function that sets flagged values to NaN for index with buffer)
16 %
17 % Jason Kelley NEWAg Lab OSU
18 % Written 29 FEB 2016
19 % Last modified 27JUL2016 (Jewell)
20
21 % check for pre-existing non-number points (errors) and interpolate during despiking
22 nn = isnan(data);
23 xs = 1:length(data);
24 if nnz(isnan(data))>0
25     data = interp1(xs(~nn),data(~nn),xs,'nearest');
26 end
27
28 w = gausswin(nw,1);           % Matlab function generates Gaussian filter
29 sw = sum(w);                 % total area under window function
30 w = w./sw;                  % normalize window
31
32 filter = conv(data,w,'same'); % filtered data using _convolution_
33
34 ii = true(length(data),1);   % eliminate edge bias: index to original data
35     hw = ceil(length(w)/2);  % data to ignore is 50% of filter window size
36     ii(1:hw) = false;
37     ii(end-hw:end) = false;
38
39 mstd = mw_std(data,nw).*sig;  % significance level in terms of standard deviation
40 fluc = zeros(length(data),1); % normalize fluctuations by absolute value
41 % fluc(ii) = (data(ii)-filter(ii))./data(ii); % alternate def for significance
```

```

42 fluc(ii) = data(ii)-filter(ii); % spikes are fluctuations exceeding signif. threshold
43 index = abs(fluc)>mstd;          % index the spikes
44 ns = nnz(index);               % count the spikes
45
46 data_ds = setnan(data,index,buffer); % set spikes and adjacent values to NaN
47 if nnz(nn)>0                    % reset pre-existing NaNs in data vector
48     data_ds(nn) = NaN;
49     data(nn) = NaN;
50 end
51 fprintf(' %i spikes removed ; ',ns)
52 fprintf('%3.3f%% of data NaN'\ed\n',(sum(isnan(data_ds))/length(data))*100)
53
54 % optional plotting for visual inspection (not included here for brevity)
55
56 % optional interpolation between nan'd points using timestamp for ordinates
57 if nargin > 4 && strcmp(varargin{1},'interp')
58     ind = isnan(data_ds);
59     switch nargin
60         case 5
61             time = 1:length(data);
62         case 6;
63             time = varargin{2};
64     end % switch
65     if nargin == 7
66         data_ds(ind) = interp1(time(~ind),data_ds(~ind),time(ind),varargin{3});
67     else
68         data_ds(ind) = interp1(time(~ind),data_ds(~ind),time(ind));
69     end
70 end % end interp option
71
72 end % end main despiking function
73
74 % sub function for moving window standard deviation
75 function mstd = mw_std(signal,w)
76     % adapted from http://matlabtricks.com/post-20/
77     % "calculate-standard-deviation-case-of-sliding-window"
78
79     N = length(signal);
80     n = conv(ones(N,1),ones(w,1),'same'); % counts no. elements in each window
81     s = conv(signal, ones(1, w), 'same'); % s vector
82     q = signal .^ 2;
83     q = conv(q, ones(1, w), 'same');      % q vector
84     mstd = (q - s.^2./n)./(n-1);          % variance of moving window
85     mstd = mstd.^0.5;                    % standard deviation
86 end % moving window mw_std sub-function
87
88 Published with MATLAB® R2016a

```

## S.2 Structure function calculation using convolution (*strfnc.m*)

```
1 function [S, max_i, fluxdir] = strfnc( trace, freq, maxlag )
2 %STRFNC Structure function calculation (following Van Atta, 1977)
3 % Last modified 09mar16
4 % INPUTS 'trace' data to analyze (Nx1 vector array)
5 % 'freq' sampling frequency (Hz)
6 % 'maxlag' maximum lag time, (seconds)
7 % OUTPUTS 'S' structure functions  $S(r)^n$  and lag  $r$  (in seconds) for rows
8 % only calculates 2nd 3rd 5th order to save memory,
9 % column order corresponds to SFs, also calculates  $-S^3(r)/r$ 
10 % format: [r  $S^2(r)$   $S^3(r)$   $-S^3(r)/r$   $S^5(r)$ ]
11 % 'max_i' relative location (iteration) at which  $S^3(r)/r$  is maximum
12 % 'fluxdir' sign of  $S^3(r)/r$ , used to determine vertical flux direction
13
14 m = length(trace);
15 lags = 1:maxlag*freq; % vector of lags from 1 to maxlag
16 rn = length(lags);
17 S = zeros(rn,5); % initialize array S to store str funcs
18
19 % method by nested iterative loops -----
20 for j = lags
21     r = lags(j); early = trace(1:end-r); later = trace(r+1:end);
22     diffs = later-early;
23     for i = [2 3 5]
24         S(j,i) = sum((diffs).^i)/(m-r);
25     end %structure functions at lag j
26 end % lags j
27 S(:,6) = lags./freq;
28
29 % method using convolution -----
30 filt = [ones(1,rn); -eye(rn)]; % singleton comparators at 1:rn lags e.g. [1 0 0 -1]
31 cT = conv2(trace,filt); % conv filter with trace to get all lags
32 cT = cT(rn+1:end-rn,:); % trim edges. 'same' does not work as with conv1.m
33 cTp(:,:,1) = power(cT,ones(m-rn,rn).*2); % for second order SF
34 cTp(:,:,2) = cTp(:,:,1).*cT; % third order SF
35 cTp(:,:,3) = cTp(:,:,1).*cTp(:,:,2); % fifth order SF
36 w = (m-rn-(1:rn))-1; % unbiased weighting vector 1/(N-1)
37 S(:,2) = sum(cTp(:,:,1),1)./w; % column order corresponds to SF order
38 S(:,3) = sum(cTp(:,:,2),1)./w; % i.e. 3rd order SF is S(:,3)
39 S(:,5) = sum(cTp(:,:,3),1)./w;
40 S(:,1) = lags./freq; % sample N -> dt
41 S(:,4) = -S(:,3)./S(:,1); % ratio used to detect lag max'ing S3r
42 % identify time lag at which  $S_3(r)/r$  is maximized, flux direction by +/-  $S^3(r)/r$ 
43 fluxdir = sign(nanmean(S(:,4),1));
44 [~,max_i] = max(fluxdir.*(S(:,4)));
45
46 end %function
47 Published with MATLAB® R2016a
```

### S.3 Cardanos Method for roots of depressed cubic polynomial (*cardanos.m*)

```
1 function [REALrts, ALLrts] = cardanos(p,q)
2 % CARDANOS(p,q) root finding algorithm for depressed cubic polynomial with real
3   valued p and q. This has reduced functionality of CardanRoots.m for limited
4   cases required for the surface renewal method. Vectors p and q (from structure
5   functions) used to determine ramp Amplitudes. polynomial should be of form  $A^3 +$ 
6    $p*A + q = 0$ , p & q real valued
7 % RETURNS REALrts: only positive real valued solutions, complex and negative
8   solutions replaced with NaN
9   ALLrts: includes positive and negatively valued and complex solutions
10  % References
11 %refs:<ahref="matlab:web('https://en.wikipedia.org/wiki/Cubic_function#Cardano.27s_me
12   thod','-browser')">Wiki</a>
13 %<ahref="matlab:web('https://www.mathworks.com/matlabcentral/newsreader/view_thre
14   ad/165013?requestedDomain=www.mathworks.com','-browser')">Source Code</a>
15
16   D = q.^2 + (4/27)*p.^3; % the discriminant
17   Dneg = D<0;
18   Dpos = ~Dneg;
19       n = size(D,1);
20       rts = zeros(n, 3); % initialization
21       a = -q(Dneg);      b = sqrt(-D(Dneg));
22       r2 = a.^2-D(Dneg);
23       rho = (4^(1/3))*exp(log(r2)/6);
24       theta = atan2(b,a)/3;
25       a = rho.*cos(theta);  b = rho.*sin(theta);
26       S1 = a;
27       x = (-0.5)*a;        y = (sqrt(3)/2)*b;
28       S2 = x-y;           S3 = x+y;
29
30   rts(Dneg,1:3) = [S1 S2 S3];
31       E = sqrt(D(Dpos));
32       u3 = (-q(Dpos)+E)/2;
33       v3 = (-q(Dpos)-E)/2;
34       u = sign(u3).*exp(log(abs(u3))/3); % Cubic roots of u3 and v3
35       v = sign(v3).*exp(log(abs(v3))/3);
36       S1 = u+v;
37       j = complex(-0.5,sqrt(3)/2); % Complex solutions
38       j2 = complex(-0.5,-sqrt(3)/2);
39       S2 = j*u+j2*v;      S3 = conj(S2);
40
41   rts(Dpos,1:3) = [S1 S2 S3];
42   ALLrts = rts;
43   rts(imag(rts)~=0) = NaN;
44   REALrts = rts;
45   end
46   Published with MATLAB® R2016a
```